HeLI-based Experiments in Swiss German Dialect Identification

Jauhiainen, Tommi Sakari

The Association for Computational Linguistics
2018-08

Jauhiainen , T S , Jauhiainen , H A & Linden , B K J 2018 , HeLI-based Experiments in þÿ S w i s s G e r m a n D i a l e c t I d e n t i f i c a t i o n . i n M Z a m p i e r i , P N a k o v , N L j u b Malmasi & A Ali (eds) , Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018) . The Association for Computational Linguistics , Santa Fe , pp. 254-262 , Workshop on NLP for Similar Languages, Varieties and Dialects , Santa Fe , United States , 20/08/2018 . < http://aclweb.org/anthology/W18-3929 >

http://hdl.handle.net/10138/308720

HeLI-based Experiments in Swiss German Dialect Identification

Tommi Jauhiainen
University of Helsinki
@helsinki.fi

Heidi Jauhiainen
University of Helsinki
@helsinki.fi

Krister Lindén
University of Helsinki
@helsinki.fi

Abstract

In this paper we present the experiments and results by the SUKI team in the German Dialect Identification shared task of the VarDial 2018 Evaluation Campaign. Our submission using HeLI with adaptive language models obtained the best results in the shared task with a macro F1-score of 0.686, which is clearly higher than the other submitted results. Without some form of unsupervised adaptation on the test set, it might not be possible to reach as high an F1-score with the level of domain difference between the datasets of the shared task. We describe the methods used in detail, as well as some additional experiments carried out during the shared task.

1 Introduction

The fifth VarDial workshop (Zampieri et al., 2018) included for the second time a shared task for German Dialect Identification (GDI). The varieties of German were from the areas of Basel, Bern, Lucerne, and Zurich. These varieties are considered dialects of Swiss German (gsw) by the ISO-639-3 standard (Lewis et al., 2013). For the first time the GDI shared task included a separate track for unknown language detection.

We have used the HeLI method and its variations in the shared tasks of the three previous VarDial workshops (Jauhiainen et al., 2015a; Jauhiainen et al., 2016; Jauhiainen et al., 2017a). The HeLI method is robust and competes with the other state-of-the-art language identification methods. For the current workshop we wanted to try out some variations and possible improvements to the original method. We submitted two different runs on the four-way classification track and in the end we did not submit any runs on the unknown language detection track.

2 Related Work

The differences between definitions of dialects and languages are not usually clearly defined, at least not in terms which would be able to help us automatically decide whether we are dealing with languages or dialects. Also the methods used for dialect identification are most of the time exactly the same as for general language identification. Language identification of close languages and dialects is one of the remaining challenges of language identification research. For a recent survey on language identification and the methods used in the field, the reader is referred to an article by Jauhiainen et al. (2018).

2.1 German dialect identification

The German dialect identification has earlier been considered by Scherrer and Rambow (2010), who used a lexicon of dialectal words. Hollenstein and Aepli (2015) experimented with a perplexity based language identifier using character trigrams. They reached an average F-score of 0.66 on sentence level between 5 German dialects.

The results of the first shared task on German dialect identification are described by Zampieri et al. (2017). Ten teams submitted results on the task utilizing a variety of machine learning methods used

Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects, pages 254–262
Santa Fe, New Mexico, USA, August 20, 2018.

for language identification. We report the results in Table 1. The methods used are listed in the first column, used features in the second column, best reached weighted F1-score in the third column, and the describing article in the fourth column.

Method	Features used	Weighted F1	Reference
Ensemble of 9 individual SVMs	char. *n*-grams 1-8 and words	0.662	(Malmasi and Zampieri, 2017)
BM25 weighted SVM	char. *n*-grams 1-5	0.661	(Bestgen, 2017)
Conditional Random Fields	char. *n*-grams, prefixes, suffixes...	0.653	(Clematide and Makarov, 2017)
SVM + SGD ensemble	*n*-grams 1-8	0.639	
Kernel Ridge Regression	*n*-grams 3-6	0.637	(Ionescu and Butnaru, 2017)
Linear SVM	char. *n*-grams and words	0.626	(Çöltekin and Rama, 2017)
Cross Entropy	char. *n*-grams up to 25 bytes	0.614	(Hanani et al., 2017)
Perplexity	words	0.612	(Gamallo et al., 2017)
Naive Bayes with TF-IDF		0.605	(Barbaresi, 2017)
LSTM NN	characters or words	0.263	

Table 1: The weighted F1-scores using different methods on the 2017 GDI test set.

2.2 Unknown language detection

Unknown languages are languages with which the language identifier has not been trained. Especially in a real-world situation it is always possible to encounter unknown languages. If the language identification method used produces a comparable score for different texts, it is possible to try thresholding. In thresholding, we find a score under (or over) which our prediction of the language is so poor, that we label it as unknown. Suzuki et al. (2002) describes a language identification method which was originally designed for identifying the language of web pages crawled from the internet. They had a predetermined threshold for each language known by the identifier and if the threshold was not reached the web page was categorized as junk. In the Finno-Ugric Languages and the Internet project (Jauhiainen et al., 2017b), we have also experimented with unknown language (or just junk) detection in order to cope with pages written in languages not known to our identifier. The method we use in production is based on a language set identification method (Jauhiainen et al., 2015b), which determines the languages used on a page. The method is simply a threshold for the number of languages: if too many languages are found in a piece of text, the text is categorized as junk. In the production environment our threshold is currently 10 languages on one web page. The production threshold has been empirically determined for the parameters used with the language set identification method and for the number of languages known by the identifier.

2.3 Language model adaptation

Language model adaptation was used by Chen and Liu (2005) for identifying the language of speech. In the system built by Chen and Liu (2005), the speech is first run through Hidden Markov Model-based phone recognizers (one for each language), which tokenize the speech into sequences of phones. The probabilities of those sequences are calculated using corresponding language models and the most probable language is selected. An adaptation routine is then used so that each of the phonetic transcriptions of the individual speech utterances is used to calculate probabilities for words t, given a word n-gram history of h as in Equation 1.

$$P_a(t|h) = \lambda P_o(t|h) + (1 - \lambda)P_n(t|h), \quad (1)$$

where P_o is the original probability calculated from the training material, P_n the probability calculated from the data being identified, and P_a the new adapted probability. λ is the weight given to original probabilities. This adaptation method resulted in decreasing the error rate in a three-way identification between Chinese, English, and Russian by 2.88% and 3.84% on an out-of-domain (different channels) data, and by 0.44% on in-domain (same channel) data.

Zhong et al. (2007) also used language model adaptation with language adaptation of speech. They evaluated three different confidence measures and the best faring measure is defined as follows:

$$C(g_i, M) = \frac{1}{n}[\log(P(M|g_i)) - \log(P(M|g_j))], \quad (2)$$

where M is the sequence to be identified, n the number of frames in the utterance, g_i the best identified language, and g_j the second best identified language. The two other evaluated confidence measures were clearly inferior. Although the $C(g_i, M)$ measure performed the best of the individual measures, a Bayesian classifier based ensemble using all the three measures gave slightly higher results. Zhong et al. (2007) use the same language adaptation method as Chen and Liu (2005), using the confidence measures to set the λ for each utterance.

3 Test setup

The dataset used in the shared task consists of manual transcriptions of speech utterances by speakers from different areas: Bern, Basel, Lucerne, and Zurich. The transcriptions are written entirely in lowercased letters. Samardžić et al. (2016) describe the ArchiMob corpus, which is the source for the shared task dataset. Zampieri et al. (2017) describe how the training and test sets were extracted from the ArchiMob corpus for the 2017 shared task. The sizes of the training and development sets can be seen in Table 2. The first track of the shared task was a standard four-way language identification between the four German dialects present in the training set.

Variety	Training	Development
Bern (BE)	32,447	8,471
Basel (BS)	30,770	11,116
Lucerne (LU)	32,955	9,966
Zurich (ZH)	32,714	9,039

Table 2: The sizes in words of the datasets distributed for the 2018 GDI shared task.

This year the GDI task also included a second track for unknown dialect detection. The unknown dialect was not included in the training or the development sets, but it was present in the test set. The test set was identical for both tracks, but the lines containing unknown dialect were ignored when calculating the scores for the first track.

4 Basic HeLI method, run 1 on track 1

We first presented the HeLI method, originally published by Jauhiainen (2010), at the VarDial 2016 (Jauhiainen et al., 2016). To make this article more self-contained, we present the full description of the method as it is used in the best submitted run for the GDI shared task. The survey by Jauhiainen et al. (2018) uses the same unified notation to define the features and methods used for language identification. This description differs from the original mostly in that we are leaving out the cut-off value c for the size of the language models as using all the available material was always the best option. When we are not using the cut-off value, no derived corpus C' consisting of the used features is generated. The final submissions were done with a system using only lowercased character 4-grams, so we present the method without the back-off function. For the complete description of the HeLI method see our VarDial 2016 article (Jauhiainen et al., 2016).

4.1 Description of the HeLI method using only 4-grams of characters

The goal is to correctly guess the language $g \in G$ for each of the lines in the test set. In the method, each language g is represented by a lowercased character 4-gram language model. The training data is tokenized into words using non-alphabetic and non-ideographic characters as delimiters and the words are lowercased. The relative frequencies of character 4-grams are calculated inside the words, so that the preceding and the following space-characters are included. The 4-grams are overlapping, so that for example a word with three characters include two character 4-grams. Then we transform the relative frequencies into scores using 10-based logarithms.

The corpus containing only the *n*-grams of the length 4 in the language models is called C^4. The domain $dom(O(C^4))$ is the set of all character *n*-grams of length 4 found in the models of any language $g \in G$. The values $v_{C_g^4}(u)$ are calculated similarly for all *n*-grams $u \in dom(O(C^4))$ for each language g, as shown in Equation 3.

$$v_{C_g^4}(u) = \begin{cases} -\log_{10}\left(\frac{c(C_g^4,u)}{l_{C_g^4}}\right) & \text{, if } c(C_g^4,u) > 0 \\ p & \text{, if } c(C_g^4,u) = 0, \end{cases} \tag{3}$$

where $c(C_g^4, u)$ is the number of *n*-grams u found in the corpus of the language g and $l_{C_g^4}$ is the total number of the *n*-grams of length 4 in the corpus of language g. These values are used when scoring the words while identifying the language of a text. The word t is split into overlapping 4-grams of characters u_i^4, where $i = 1, ..., l_t - 4$. l_t is the length of the word in characters, including the preceding and the following space-characters. Each of the *n*-grams u_i^4 is then scored separately for each language g.

If the *n*-gram u_i^4 is found in $dom(O(C_g^4))$, the values in the models are used. If the *n*-gram u_i^4 is not found in any of the models, it is simply discarded. We define the function $d_g(t, 4)$ for counting *n*-grams in t found in a model in Equation 4.

$$d_g(t,4) = \sum_{i=1}^{l_t-4} \begin{cases} 1 & \text{, if } u_i^4 \in dom(O(C^4)) \\ 0 & \text{, otherwise.} \end{cases} \tag{4}$$

When all the *n*-grams of the size 4 in the word t have been processed, the word gets the value of the average of the scored *n*-grams u_i^4 for each language, as in Equation 5.

$$v_g(t,4) = \frac{1}{d_g(t,4)} \sum_{i=1}^{l_t-4} v_{C_g^4}(u_i^4) \quad \text{, if } d_g(t,4) > 0, \tag{5}$$

where $d_g(t, 4)$ is the number of *n*-grams u_i^4 found in the domain $dom(O(C_g^4))$. If all of the *n*-grams of the size 4 were discarded, $d_g(t, 4) = 0$, a word gets the penalty value p for every language, as in Equation 6.

$$v_g(t, 0) = p \tag{6}$$

The mystery text is tokenized into words using the non-alphabetic and non-ideographic characters as delimiters. The words are lowercased. After this, a score $v_g(t, 4)$ is calculated for each word t in the mystery text for each language g. The whole line M gets the score $R_g(M)$ equal to the average of the scores of the words $v_g(t, 4)$ for each language g, as in Equation 7.

$$R_g(M) = \frac{\sum_{i=1}^{l_{T(M)}} v_g(t_i, 4)}{l_{T(M)}} \tag{7}$$

where $T(M)$ is the sequence of words and $l_{T(M)}$ is the number of words in the line M. Since we are using negative logarithms of probabilities, the language having the lowest score is returned as the language with the maximum probability for the mystery text.

4.2 Experiments on the development set and results on the test set

The training dataset was completely written in lowercase so we used only lowercased versions of the language models. First we tested the effect of not using all the data in the language models with varying the cut-off parameter c on the development set. The largest language model was the character 6-gram model for the Basel-area dialect with 16,947 different 6-grams. The results using optimized penalty values for each c are presented in Table 3. The results would seem to indicate that the recall starts to decline in an increasing manner as soon as some of the material from the language models is left out. This

is something we have noticed before in settings where the training data is of very good quality. If we are using the identifier in a production setting with, for example, Wikipedia-derived language models, some of the models include so much noice that not using the complete models improves the results (Jauhiainen et al., 2017b). The optimal penalty value is also clearly tied to the maximum size of the used language models.

Lowercased words	**Lowercased** n_{max}	**Penalty** p	**Cut-off** c	**Recall**
yes	8	5.1	16,947	64.47%
yes	8	4.9	15,000	64.41%
yes	8	4.8	10,000	64.19%

Table 3: Basic HeLI method results on the development set with varying c.

We decided to use all the available data and then optimized the used language models and the penalty value p. The results on the development set with different model combinations can be seen in Table 4. The penalty value p presented in the third column was the optimal one for each configuration. The HeLI method using character *n*-grams from one to four attained the best recall 65.97%.

Lowercased words	**Lowercased** n_{max}	**Penalty** p	**Recall**
no	4	5.8	65.97%
no	5	5.4	65.39%
no	6	5.3	64.79%
no	8	5.4	64.56%
no	7	5.1	64.51%
yes	8	5.1	64.47%
no	3	5.7	62.80%
no	2	5.7	53.43%
no	1	5.6	33.13%

Table 4: Basic HeLI method results on the development set using different language model combinations.

We also experimented with leaving out the lower order *n*-grams. The results of these experiments on the development set can be seen in Table 5. To our surprise, the best results were attained using only the 4-grams of characters, which means that the backoff function of the HeLI method is not used at all. The recall on the development set was 66.10%. We also re-tested using lower cut-offs c, but leaving off any material in the language models only made the results worse again.

Lowercased words	**Lowercased** n**-gram range**	**Penalty** p	**Recall**
not used	4 - 4	5.8	66.10%
not used	1 - 4	5.8	65.97%
not used	2 - 4	5.8	65.97%
not used	3 - 4	5.8	65.97%

Table 5: Basic HeLI method results on the development set with different *n*-gram ranges.

We decided to use the character 4-grams and the penalty value of 5.8 for the first run. We added the development data to the training data and generated new models. The system attained a recall of 63.97% on the test set, which was somewhat less than what we had seen with the development set.

5 HeLI with language model adaptation, run 2 on track 1

While experimenting with the basic HeLI method we created a test setting to detect the difference of using out-of-domain and in-domain training data. For each language, we divided the development set in two halves. We experimented with adding the first half of the development data (we call it *dv-dv*) to the training data (*tr*) of each language, creating new language models and testing them on the second half of the development data (*dv-tst*). The recalls on the second half of the development set (*dv-tst*) using the combined *tr* and *dv-dv* for training were much better than the recalls on the first half (*dv-dv*) using just *tr* for training. The recalls can be seen in the fourth column of Table 6. In the heading of the table, the training data used is indicated in parenthesis after the test data. We decided to try to test *dv-tst* also

without including *dv-dv* in the training set in order to see if *dv-tst* was for some reason generally easier to identify than *dv-dv*. The second half of the development data, *dv-tst*, turned out to be a little bit easier to identify than the first half, *dv-dv*, using the original models generated from just the training data as can be seen in the fifth column of Table 6. Another hypothesis is that the development data is from an another source than the training data and the first half introduces a great number of new words which are relevant to the second half. Also we wanted to know if just 15.9% increase in training text amount could generate this much better recall. In order to test this hypothesis we removed the same amount of lines as was in *dv-dv* from the training data, marked as *tr-sz(dv-dv)* in the table, and inserted the *dv-dv* lines instead. The recall percentages from those tests are in the final column of Table 6 and they suggest that the development data is indeed from a somewhat different domain than the training data and the identifier actually performs better when some of the original training data is removed.

It can also be seen from the results of the experiments that the best models for in-domain experiments were word and character *n*-grams from one to five and for the out-of-domain they were character *n*-grams from one to four or just 4-grams. This would then indicate that if the domain of the language to be tested is the same or similar to the one that the models have been created from, the models could use longer character *n*-grams and words, if not, then using just character *n*-grams is a better strategy.

***n*-gram range**	**Words**	**dv-dv (tr)**	**dv-tst (tr+dv-dv)**	**dv-tst (tr)**	**dv-tst (tr-sz(dv-dv)+dv-dv)**
1 - 8	yes	63.65%	78.65%	65.46%	79.55%
1 - 7	yes	63.82%	78.95%	65.89%	79.68%
1 - 6	yes	64.16%	78.91%	66.02%	79.90%
1 - 5	yes	64.59%	79.55%	65.98%	80.41%
1 - 4	yes	64.76%	79.25%	66.07%	79.94%
1 - 3	yes	65.28%	78.74%	66.67%	79.77%
1 - 2	yes	65.11%	78.35%	66.37%	79.17%
1	yes	63.95%	78.01%	65.29%	78.57%
-	yes	63.61%	77.53%	64.86%	78.01%
1 - 8	no	63.65%	78.74%	65.59%	79.47%
1 - 7	no	63.73%	78.91%	65.42%	79.51%
1 - 6	no	64.21%	78.95%	65.21%	79.55%
1 - 5	no	65.15%	79.21%	65.76%	79.94%
1 - 4	no	65.45%	78.35%	66.58%	79.25%
4	no	65.28%	78.14%	67.35%	78.57%
1 - 3	no	61.89%	73.20%	63.70%	73.97%
1 - 2	no	52.88%	60.87%	54.08%	61.60%
1	no	33.18%	36.34%	33.08%	37.5%

Table 6: Baseline HeLI recalls using different combinations of training and development sets.

What we learned from these experiments with the basic HeLI method is that, if we would be able to somehow incorporate well identified sentences into the original models it might introduce crucial new word or character *n*-gram vocabulary. We decided to try always adding the character 4-grams from the most confidently identified sentence to the language model of the respective language and re-identifying the rest, always marking the best identified sentence as not needing to be identified again. This process is recursive and it runs until all the sentences except the last one are used for language modelling. In order to decide which sentence is most confidently identified, we need a confidence score. As a confidence measure CM, we used the difference between the scores of the best $R_g(M)$ and the second best $R_h(M)$ identified language for each line. Later we found that basically the same confidence measure was earlier proposed by Zhong et al. (2007). In our case it is calculated using the Equation 8:

$$CM(C_g, M) = R_h(M) - R_g(M) \tag{8}$$

where M is the line containing the mystery text. It could be beneficial to end the recursive adaptation before all the sentences are exhausted, if the confidence score is reliable enough. However, we did not have time to experiment with a cut-off value for the confidence score before the submissions were due.

The identifier with language model adaptation reached 77.99% recall on the development set with the same language models (character 4-gram) and penalty value (5.8) which we used with the basic HeLI method in run 1. It was an increase of 12.71% on top of the recall of the basic HeLI method.

We suspected that using higher order *n*-grams or words could produce even better results, but we did not have time to test this theory. We added the development data to the training data, generated new language models and submitted our run 2. The second run reached a recall of 69.19% on the test set, an increase of 5.22%. The macro F1-score attained on the run 2 was 0.6857. The results are very good considering that there was an unknown dialect within the actual test set and all the lines in the unknown dialect were incorrectly incorporated into some of the language models. The final results compared with the best results submitted by other teams are shown in Table 7.

System	F1 (macro)
HeLI with adaptive language models, run 2	**0.6857**
benf	0.6464
safina	0.6449
taraka_rama	0.6398
The basic HeLI method, run 1	**0.6386**
LaMa	0.6374
XAC	0.6336
GDI_classification	0.6203
dkosmajac	0.5909
Random Baseline	0.2521

Table 7: Results compared with the other submitted runs. Our submitted results are bolded.

6 Experiments with unknown language detection, track 2

The basic HeLI method always maps the mystery text M into one of the languages it has been trained with. The 2015 Discriminating Between Similar Languages shared task included an unknown category which contained several a priori unknown languages. One of the methods we used in 2015 was using a threshold for the score $R_g(M)$ to detect the unknown language. In order to assess the suitability of using the threshold score with the German dialects, we compared the range of the scores when g was correctly or incorrectly identified using the character 4-gram language models on the development set. The score ranges can be seen in Table 8, where the line with correct identifications is bolded. The lower the score, the better the mystery text fits the language. The scores ranged from 1.28 to 4.56 when the dialect was correctly identified, with most of the scores higher than the lower ranges of the incorrect identifications. The fact that the worst absolute score (4.56) was attained with a correct identification drove us to the conclusion that simply using the score as a cut-off would not be a quick solution to the unseen language problem. Due to time restrictions, we did not pursue this investigation further. We were also unable to test the language set based thresholding method we are using in the production environment. In the end, we did not submit any results to the unknown language detection track.

Correct language	Identified language	Lowest score	Highest score
ZH	**ZH**	**1.28**	**4.56**
ZH	LU	2.37	4.23
ZH	BS	1.71	4.21
ZH	BE	2.26	3.95

Table 8: Score ranges when trying to identify the dialect from Zurich area.

7 Conclusions

The macro F1-score attained by the basic HeLI method is within 0.0078 score difference to the best five results submitted by the other teams. Unsupervised language model adaptation improved on the recall of the basic HeLI-method by 5.22%. The score difference between our run using the adaptive language models and the second best submitted run is 0.0393. Language model adaptation would seem to be especially usable in situations where the training material can be expected to be from a different domain than the material to be identified. The adaptation method proved to be very robust as it performed well even with the unknown language present in the test set.

Acknowledgments

This research was partly conducted with funding from the Kone Foundation Language Programme (Kone Foundation, 2012).

References

Adrien Barbaresi. 2017. Discriminating between Similar Languages using Weighted Subword Features. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, pages 184–189, Valencia, Spain.

Yves Bestgen. 2017. Improving the Character Ngram Model for the DSL Task with BM25 Weighting and Less Frequently Used Feature Sets. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, pages 115–123, Valencia, Spain.

Çağri Çöltekin and Taraka Rama. 2017. Tübingen system in VarDial 2017 shared task: experiments with language identification and cross-lingual parsing. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, pages 146–155, Valencia, Spain.

Yingna Chen and Jia Liu. 2005. Language Model Adaptation and Confidence Measure for Robust Language Identification. In *Proceedings of International Symposium on Communications and Information Technologies 2005 (ISCIT 2005)*, volume 1, pages 270–273, Beijing, China.

Simon Clematide and Peter Makarov. 2017. CLUZH at VarDial GDI 2017: Testing a Variety of Machine Learning Tools for the Classification of Swiss German Dialects. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, pages 170–177, Valencia, Spain.

Pablo Gamallo, Jose Ramom Pichel, and Iñaki Alegria. 2017. A Perplexity-Based Method for Similar Languages Discrimination. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, pages 109–114, Valencia, Spain.

Abualsoud Hanani, Aziz Qaroush, and Stephen Taylor. 2017. Identifying dialects with textual and acoustic cues. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, pages 93–101, Valencia, Spain.

Nora Hollenstein and Noëmi Aepli. 2015. A Resource for Natural Language Processing of Swiss German Dialects. In *Proceedings of the International Conference of the German Society for Computational Linguistics and Language Technology (GSCL 2015)*, pages 108–109, University of Duisburg-Essen, Germany.

Radu Tudor Ionescu and Andrei Butnaru. 2017. Learning to Identify Arabic and German Dialects using Multiple Kernels. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, pages 200–209, Valencia, Spain.

Tommi Jauhiainen, Heidi Jauhiainen, and Krister Lindén. 2015a. Discriminating Similar Languages with Token-Based Backoff. In *Proceedings of the Joint Workshop on Language Technology for Closely Related Languages, Varieties and Dialects (LT4VarDial)*, pages 44–51, Hissar, Bulgaria.

Tommi Jauhiainen, Krister Lindén, and Heidi Jauhiainen. 2015b. Language Set Identification in Noisy Synthetic Multilingual Documents. In *Proceedings of the Computational Linguistics and Intelligent Text Processing 16th International Conference, CICLing 2015*, pages 633–643, Cairo, Egypt.

Tommi Jauhiainen, Krister Lindén, and Heidi Jauhiainen. 2016. HeLI, a Word-Based Backoff Method for Language Identification. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial3)*, pages 153–162, Osaka, Japan.

Tommi Jauhiainen, Krister Lindén, and Heidi Jauhiainen. 2017a. Evaluating HeLI with Non-Linear Mappings. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, pages 102–108, Valencia, Spain.

Tommi Jauhiainen, Krister Lindén, and Heidi Jauhiainen. 2017b. Evaluation of Language Identification Methods Using 285 Languages. In *Proceedings of the 21st Nordic Conference on Computational Linguistics (NoDaLiDa 2017)*, pages 183–191, Gothenburg, Sweden. Linköping University Electronic Press.

Tommi Jauhiainen, Marco Lui, Marcos Zampieri, Timothy Baldwin, and Krister Lindén. 2018. Automatic Language Identification in Texts: A Survey. *arXiv preprint arXiv:1804.08186*.

Tommi Jauhiainen. 2010. Tekstin kielen automaattinen tunnistaminen. Master's thesis, University of Helsinki, Helsinki.

Kone Foundation. 2012. The Language Programme 2012-2016. http://www.koneensaatio.fi/en.

M. Paul Lewis, Gary F. Simons, and Charles D. Fennig, editors. 2013. *Ethnologue: Languages of the world, seventeenth edition*. SIL International, Dallas, Texas.

Shervin Malmasi and Marcos Zampieri. 2017. German Dialect Identification in Interview Transcriptions. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, pages 164–169, Valencia, Spain.

Tanja Samardžić, Yves Scherrer, and Elvira Glaser. 2016. ArchiMob–A corpus of spoken Swiss German. In *Proceedings of the Language Resources and Evaluation (LREC)*, pages 4061–4066, Portoroz, Slovenia.

Yves Scherrer and Owen Rambow. 2010. Word-based Dialect Identification with Georeferenced Rules. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP 2010)*, pages 1151–1161, Massachusetts, USA. Association for Computational Linguistics.

Izumi Suzuki, Yoshiki Mikami, Ario Ohsato, and Yoshihide Chubachi. 2002. A Language and Character Set Determination Method Based on ngram Statistics. *ACM Transactions on Asian Language Information Processing (TALIP)*, 1(3):269–278.

Marcos Zampieri, Shervin Malmasi, Nikola Ljubešić, Preslav Nakov, Ahmed Ali, Jörg Tiedemann, Yves Scherrer, and Noëmi Aepli. 2017. Findings of the VarDial Evaluation Campaign 2017. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, Valencia, Spain.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Ahmed Ali, Suwon Shon, James Glass, Yves Scherrer, Tanja Samardžić, Nikola Ljubešić, Jörg Tiedemann, Chris van der Lee, Stefan Grondelaers, Nelleke Oostdijk, Antal van den Bosch, Ritesh Kumar, Bornini Lahiri, and Mayank Jain. 2018. Language Identification and Morphosyntactic Tagging: The Second VarDial Evaluation Campaign. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, Santa Fe, USA.

Shan Zhong, Yingna Chen, Chunyi Zhu, and Jia Liu. 2007. Confidence measure based incremental adaptation for online language identification. In *Proceedings of International Conference on Human-Computer Interaction (HCI 2007)*, pages 535–543, Beijing, China.